REMARKS

Reconsideration of the above-identified application in view of the following remarks is respectfully requested. No new material has been entered, and no amendments have been made.

Claims

Claims 1 – 9, 11, 13 – 24, 26, and 28 – 33 are in this case.

The Examiner has rejected the claims under 35 USC §103(a) as being unpatentable over Rosen (US 6081790) further in view of Drummond (US 6289320) further in view of Menezes (*Handbook of Applied Cryptography*) further in view of Cooke (US 6675153) further in view of Muller (*A Survey of Programming Techniques*) and further in view of Gong (*Inside Java 2 Platform Security*). The Examiner has additionally rejected some of the claims further in view of Pratt (US 6070254) further in view of Sommerer (*The Java Archive JAR File Format*) and further in view of Wright (*Dynamic Data Structures*).

The Applicant respectfully traverses the above rejections, as discussed in detail below.

Office Action Has Not Established the Required *Prima Facie* Case of Obviousness

The Applicant respectfully maintains that the present Office Action has not established the required *prima facie* case of obviousness to sustain the 35 USC §103(a) rejections against the present invention.[1] In particular, there are three basic criteria necessary for establishing a *prima facie* case, all of which are necessary[2], but none of these three criteria are met by the present Office Action. This is summarized below:

[1] MPEP § 2142 states: "The examiner bears the initial burden of factually supporting any *prima facie* conclusion of obviousness. If the examiner does not produce a *prima facie* case, the applicant is under no obligation to submit evidence of nonobviousness."

[2] MPEP § 2143 states: "To establish a *prima facie* case of obviousness, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge

As is shown herein:

1. *There is no suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings, as required.*

   It is stated in MPEP § 2143, and elsewhere, that a reasonable motivation or desirability of combining and modifying prior art references must accompany the proposed combination or modification.[3, 4, 5, 6]

- The prior art lacks any suggestion that the cited references should be, or could be, combined and/or modified to meet the present claims. The cited references themselves are from different fields and do not teach or suggest combining them or modifying them in the fashion proposed by the present Office Action.

- The basic suggestions for combining references, and the proposed motivations in the present Office Action are vague and indefinite, and do not explain with reasonable specificity how the references should be combined and modified, and why it would be desirable to do so. The Office Action therefore does not procedurally establish a *prima facie* case of obviousness.[6]

- Certain teachings of the cited references are misunderstood by the present Office Action and do not support what the Office Action purports to establish as a motivation or as a result of the combination.

generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations."

These three basic necessary criteria are also stated elsewhere in MPEP, including MPEP § 706.2(j) and MPEP § 2143. In addition, MPEP § 2141 contains further tenets of patent law as basic considerations.

[3] MPEP § 2143.01 states: "The mere fact that references can be combined or modified is not sufficient to establish *prima facie* obviousness." — *In re Mills,* 916, F. 2d 680, 1.6 USPQ2d 1430 (Fed. Cir. 1990)

[4] MPEP § 2142 states: "When the motivation to combine the teachings of the references is not immediately apparent, it is the duty of the examiner to explain why the combination of the teachings is proper." — *Ex parte Skinner*, 2 USPQ2d 1788 (Bd. Pat. App. & Inter. 1986)

[5] MPEP § 706.2(j) states: "The initial burden is on the examiner to provide some suggestion of the desirability of doing what the inventor has done. 'To support the conclusion that the claimed invention is directed to obvious subject matter, either the references must expressly or impliedly suggest the claimed invention or the examiner must present a convincing line of reasoning as to why the artisan would have found the claimed invention to have been obvious in light of the teachings of the references.' " — *Ex parte Clapp*; 227 USPQ 972,973 (Bd. Pat. App. & Inter. 1985)

[6] MPEP § 2142 states: "A statement of a rejection that includes a large number of rejections must explain with reasonable specificity at least one rejection, otherwise the examiner procedurally fails to establish a *prima facie* case of obviousness." — *Ex parte Blanc*, 13 USPQ2d 1383 (Bd. Pat. App. & Inter. 1989)

- Certain motivations proposed in the present Office Action for combining the references are not reasonable in view of the prior art.

- Certain combinations of references and proposed motivations for doing so, as stated in the present Office Action, are inconsistent and contradict one another, thus showing that the Office Action is not considering the present invention as a whole, nor the cited prior art references as a whole.[7]

2. *There is no reasonable expectation of success in combining and modifying the references, as required.*[2]

- The cited prior-art systems are from diverse unrelated fields, and are from fields unrelated to that of the present invention. Elements taken from one prior-art system are incompatible, both in their structure and in their intended purposes, with the elements of different prior-art systems, and cannot reasonably be expected to achieve coherent structural result of the same form as the present invention exhibits.

- Moreover, the cited prior art systems are already complete and functional as presented in the prior art, and would not benefit from combination and modification as proposed by the present Office Action.

- Furthermore, prior art evidence furnished by those highly-skilled in the art indicates that modifying and combining certain cited references by those who are ordinarily-skilled in the art (as proposed by the Office Action) would not be successful. The prior art thus teaches away from the combining and modifying proposed by the Office Action.

3. *The proposed combination of references does not teach or suggest all the claim limitations, as required.*[2, 8]

- Individually and in combination, the references cited in the Office Action do not meet the claims of the present application.

[7] MPEP § 2141 states: "When applying 35 U.S.C. 103, the following tenets of patent law must be adhered to: (A) The claimed invention must be considered as a whole; (B) The references must be considered as a whole and must suggest the desirability and thus the obviousness of making the combination; (C) The references must be viewed without the benefit of impermissible hindsight vision afforded by the claimed invention; and (D) Reasonable expectation of success is the standard with which obviousness is determined. *Hodosh v. Block Drug Co., Inc.*, 786 F.2d 1136, 1143 n.5, 229 USPQ 182, 187 n.5 (Fed. Cir. 1986).

[8] In addition to MPEP § 2143, MPEP § 2142 also states that to establish a *prima facie* case of obviousness, "the prior art reference (or references when combined) must teach or suggest all the claim limitations."

- Claim limitations which are not met by the combination of references proposed by the Office Action include significant limitations which represent the present invention, and which appear in the independent claims of the present application.

In addition, the present invention materially teaches away from the prior art, and this should be considered as a significant factor in establishing non-obviousness of the present invention.

The above points are covered in detail below.

Detailed Remarks

Following is a detailed point-by-point analysis of the prior art references and the combinations and modifications thereof as put forth in the present Office Action along with the motivations proposed by the Office Action as a case of obviousness against the present invention.

1. Lack of Convincing Line of Reasoning and Reasonable Specificity in Motivations for Proposed Combinations

As required by MPEP, the Office Action "must present a convincing line of reasoning as to why the artisan would have found the claimed invention to have been obvious in light of the teachings of the references,"[5] and the rejections must be explained with "reasonable specificity" in order to procedurally establish a *prima facie* case of obviousness.[6] The Applicant respectfully traverses the rejections, in that the present Office Action exhibits neither a convincing line of reasoning nor reasonable specificity, as can be seen below:

The proposed motivations are an important issue, because reasonable and credible motivations are essential elements of a *prima facie* case for an obviousness rejection.[3, 4] The motivations proposed in the present Office Action are listed verbatim, in their entirety, as follows (some of these are further considered individually, below):

Proposed motivation (verbatim, in the entirety)	Claim(s)	page(s)
"allow operation"	1, 24, 28	3
"maintain data integrity"	1, 24, 28	3
"allow printing of a receipt"	1, 24, 28	4
"enable usage of general procedures in other programs"	1, 24, 28	4
"protect the user"	1, 24, 28	5
"allow memory to be allocated at execution time"	6, 7	7
"JAR format allows for compression"	18, 26	8, 9
"determine if an error has been detected which requires modification of the data"	13, 14	9

None of these proposed motivations exhibit a "convincing line of reasoning"[5] or a "reasonable specificity"[6], as shown below. In general, the proposed motivations are vague and indefinite.

- In some cases, in place of a reasonable motivation, the Office Action merely recites a well-known feature inherent in the reference being combined. This does not fairly constitute a "convincing line of reasoning" nor does it constitute "reasonable specificity".
  - For example, "allow memory to be allocated at execution time" is proposed as a motivation to combine the dynamic memory allocation teachings of Wright. To "allow memory to be allocated at execution time" is merely one of the well-known features of dynamic memory allocation. It is not by itself a reasonable motivation for combining the reference, however, because *dynamic memory allocation is not always desirable or appropriate:* it is well-known in the art that there are cases where static memory allocation (where memory is pre-allocated before execution time) is preferable or necessary. (Also, see below for analysis of the misunderstanding of the Wright reference by the Office Action.)
  - Likewise, "JAR format allows for compression" simply by itself does not serve as a reasonable motivation for combining the reference, because (as is also well-known in the art) *data compression is not always desirable or feasible.*
  - Even "determine if an error has been detected" by itself is not a reasonable motivation to combine. As shown in detail further below, it is well-known in the

art that *the error-detecting properties of the secondary reference are already inherent to a greater and more useful degree in the primary references*. Thus, mere error-determination is not a reasonable motivation to combine them. Furthermore, "modification of the data" is vague and indefinite. What kind of "modification" should be made? Why should it be made? How should it be made? Is "modification" supposed to mean "correction"? These points are not addressed by the Office Action. (Also, see below for analysis of the misunderstanding of the Pratt reference by the Office Action.)

- **Office Action Lacks Reasonable Specificity regarding Rosen-Drummond**: In the §103(a) rejection of claims 1, 24, and 28 (Office Action page 3), the Office Action states that "Rosen fails to disclose using the personalization to authorize use. However Drummond teaches such authorization ... it would have been obvious to a person of ordinary skill in the art to use the personalization of Rosen for authorization. Motivation to do so would have been to allow operation."

The Applicant respectfully traverses this rejection in that the proposed motivation lacks reasonable specificity (allow operation of *what*? — and *how* is operation allowed?). Rosen describes a system for presentation and payment of invoices over a network, and is fully operational as disclosed. Drummond describes an automated banking machine which already has mechanisms in place to allow operation (Drummond col. 14 lines 19 - 38). *If both the primary and secondary references already allow operation, it is not necessary to combine them "to allow operation" — so what is the desirability of making such a combination?* These points are not addressed in the Office Action.

Based on the content of the cited excerpts of Rosen (column 4, lines 29 - 35) and Drummond (Drummond col. 14 lines 19 - 38), it may be inferred that the Office Action proposes the combination to allow operation of Drummond's banking machine. If so, however, the proposed use of Rosen's presentment ticket containing customer information to perform Drummond's authorization to allow use of a banking machine would change the principle of operation of

Drummond's system[9] and would render Drummond's system unsatisfactory for its intended purpose[10], as follows:

Drummond already provides for authorization via a digital signature. In Drummond's scheme, this digital signature is associated with an HTTP server and is received responsive to a Java script which is subsequently processed in a device application portion. The digital signature is processed, and if valid (such as matching a stored digital signature), then banking machine operation is authorized (Drummond col. 14 lines 19 - 36). It is clear from Drummond's description of the authorization process that the operation of the entire system depends on a complex interrelationship among the various components, including the digital signature. For example, Drummond discloses a diverse set of customer verification data including not only PIN and biometric information but also specific confidential banking data for inclusion in the authorization process (Drummond, col. 13 lines 4 - 32). In contrast to Drummond's extensive and persistent customer verification data (required for long-term banking machine and bank account access), Rosen's customer data is sparse and may be transient (required for a short-term single transaction only). The suggestion which can be inferred from the Office Action is that the limited customer data in Rosen's presentment ticket (Rosen col. 4 lines 29 - 35) could be used for authorization instead of Drummond's digital signature. But this would change the principle of operation of Drummond, and therefore the teachings of the references are not sufficient to render the subject claims *prima facie* obvious.[9]

Furthermore, *this modification would also render Drummond unsatisfactory for its intended purpose,*[10] in that authorization to use the banking machine would be less stringent and would become lax. For example, a banking machine authorization based on Rosen's limited and transient customer data would not contain confidential bank account history and balance information, and would therefore dispense cash to a customer whose account was badly overdrawn.

[9] MPEP § 2143.01 states: "The proposed modification cannot change the principle of operation of a reference — If the proposed modification or combination of the prior art would change the principle of operation of the prior art invention being modified, then the teachings of the references are not sufficient to render the claims *prima facie* obvious." *In re Ratti*, 270 F. 2d 810, 123 USPQ 349 (CCPA 1959)

[10] MPEP § 2143.01 states: "The proposed modification cannot render the prior art unsatisfactory for its intended purpose — If proposed modification would render the prior art invention being modified unsatisfactory for its intended purpose, then there is no suggestion or motivation to make the proposed modification." *In re Gordon*, 733 F. 2d 900, 221 USPQ 1125 (Fed. Cir. 1984)

Drummond provides for this contingency (Drummond, col. 13, lines 24 - 27), *but Rosen does not.* Thus, the Office Action's suggestion and proposed motivation for combining Rosen and Drummond ("allow operation") are not valid for showing obviousness.[10]

- **Office Action Lacks Reasonable Specificity regarding Rosen-Gong**: In the §103(a) rejection of claims 1, 24, and 28 by combining with Gong (Office Action pages 4 - 5) the Office Action states that "The modified Rosen, Drummond, Menezes, Cook, and Muller system fails to disclose the information stream is associated with a deliverable published software. However, Gong teaches such software (see pages 23 - 25). ... it would have been obvious to a person of ordinary skill in the art to apply the personalization of the modified Rosen, Drummond, Menezes, Cook, and Muller system to the deliverable published software of Gong. Motivation to do so would have been to protect the user (see pages 23 - 25)."

  The Applicant respectfully traverses the rejection in that the proposed motivation "to protect the user" lacks reasonable specificity. How does applying the personalization of Rosen to Gong's software "protect the user"? What does it "protect the user" from? Gong (pages 23 - 25) describes the basic Java security architecture, which Gong notes already offers protection to the user (for example, page 24 "... if a user accidentally imports a hostile applet, that applet cannot damage the user's system."). Gong therefore does not need Rosen's personalization to protect the user, at least not from hostile applets. Rosen's personalization is not associated with software (as noted by the Office Action itself), so hostile applets do not threaten a system based on Rosen; and therefore Rosen cannot benefit from Gong for protection. The Office Action, however, does not present any explanation of what the alleged user protection is, how it would work, or why it would be desirable to combine these references.

  Based on the content of the cited excerpts of Rosen (column 4, lines 29 - 35), Drummond (Drummond col. 14 lines 19 - 38), and Gong (pages 23 - 25) it might be inferred that the Office Action proposes the combination to use Rosen's presentment ticket containing customer information to perform Drummond's authorization of a banking machine to attain user protection in the environment of Gong's software. If so, however, this would change the principle of operation of Gong's software,[9] which is currently based on Java security architecture, to something else, and would render Gong's software unsatisfactory for its intended purpose.[10]

It is also noted that the proposed motivation for making this combination with Gong is inconsistent with the earlier proposed motivation for combining Rosen and Drummond (above). The proposed motivation for combining Rosen and Drummond was to "allow operation", presumably of Drummond's banking machine that the user wanted to access. The proposed motivation for the combination with Gong is presumably to protect the user in connection with the use of Gong's published software that the user wants to use. These two proposed motivations are disparate and incompatible, and are not unified by any inventive concept.

This disparity and inconsistency in the proposed motivations further shows that there is no *prima facie* case of obviousness, because the present Office Action lacks consideration of the invention as a whole and of the references as a whole.[7]

- **Office Action is Inconsistent and Proposes Contradictory Motivations for Combination**: As further indication that the present Office Action lacks consideration of the invention as a whole and of the references as a whole, it is noted that there are inconsistencies and contradictions in the Office Action.

  - On page 3 of the present Office Action, it is stated that "As per claims 1, 24, and 28, Rosen discloses a personalization ... incorporated into the information stream and prior to <u>receipt of the published software by the customer</u>..." On page 4, however, the Office Action states that "The modified Rosen, Drummond, Mènezes, Cook, and Muller system <u>fails to disclose the information stream is associated with deliverable published software</u>." (emphasis supplied)

    If, as stated on page 4, Rosen, Drummond, Menezes, Cook, and Muller fail to disclose deliverable published software, then the statement on page 3 is misrepresentative of Rosen. Therefore, this contradiction in the Office Action shows that Rosen-Drummond does <u>*not*</u> establish the obviousness of claims containing a limitation of deliverable published software, as in the present claims 1, 24, and 28.

    It can be seen that none of the cited prior art references disclose anything related to deliverable published software, and the Applicant therefore respectfully traverses the rejection on page 3 of the present Office Action of claims 1, 24, and 28. This is covered in further detail below.

- In another contradiction, the present Office Action proposes (page 3) that a motivation for combining Rosen and Drummond is to "allow operation". On page 5, however, the Office Action states that "the modified Rosen, Drummond, Menezes, Cook, Muller, Gong system discloses the personalization has no usage restriction associated with it". (emphasis supplied) According to the controlling definition[11] put forth in the present application (page 39 lines 5 - 10), however, *any feature that is inserted in software to "allow operation" constitutes a form of usage control and thus represents a "usage restriction".*

- **Office Action Lacks Reasonable Specificity regarding Rosen-Drummond-Menezes**: In the §103(a) rejection of claims 1, 24, and 28 by combining with Menezes (Office Action page 3), the Office Action states that "The modified Rosen and Drummond system fails to disclose authenticating the personalization data. However, Menezes teaches such authentication (see page 361) ... it would have been obvious to a person of ordinary skill in the art to authenticate the personalization data of Rosen. Motivation to do so would have been to maintain data integrity (see page 361)."

  The Applicant respectfully traverses this rejection, in that the proposed motivation "to maintain data integrity" lacks reasonable specificity. In particular, *both Rosen and Drummond already have provisions to "maintain data integrity"* by utilizing Internet technology to assure data integrity (see below for details). Therefore, it is necessary to explain with reasonable specificity any proposed motivation for making further provisions. Otherwise, why would someone ordinarily skilled in the art be motivated to combine and modify certain prior art systems to do what those prior art system already do? These points are not addressed in the Office Action.

  Internet technology (see Socolofsky, *RFC 1180: A TCP/IP Tutorial*, page 2) is used both by Rosen (col. 1 lines 47 - 49) and by Drummond (col. 2 lines 40 - 42) and therefore *Rosen and Drummond already maintain data integrity.* As is well-known

[11] It is stated in MPEP § 2111.01 (III.) and elsewhere that "An applicant is entitled to be his or her own lexicographer and may rebut the presumption that claim terms are to be given their ordinary and customary meaning by clearly setting forth a definition of the term that is different from its ordinary and customary meaning(s)... Where an explicit definition is provided by the applicant for a term, that definition will control interpretation of the term as it is used in the claim".

in the art, TCP is a reliable protocol that uses checksums to verify data integrity and handshaking to guarantee that transmitted data is received intact (see Ahonen, *Transport Control Protocol,* pages 5, 7 - 13).

Based on the lack of reasonable specificity, as described above, the Office Action procedurally does not establish a *prima facie* case of obviousness.[6]

Furthermore, teachings of certain cited references are misunderstood by the present Office Action. Where the cited references are misunderstood, there is not only lack of a convincing line of reasoning for combining the references, but there also cannot be a reasonable expectation for success.

- **Office Action Misunderstands Wright**: In the §103(a) rejection of claims 6 - 7 (Office Action page 7), the Office Action states (page 7) that "the modified Rosen, Drummond, Menezes, Cook, Muller, Gong system fails to disclose that the personalization does not have a fixed address and extent within the information stream... However, Wright teaches such limitations (see pages 2 - 3)... it would have been obvious to a person of ordinary skill in the art to use Wright's method of dynamically allocating memory for the personalization of the Rosen, Drummond, Menezes, Cook, Muller, Gong system ... Motivation to do so would have been to allow memory to be allocated at execution time (see page 2)."
    - The Applicant respectfully traverses the above contention, in that the Office Action apparently misunderstands the nature of the subject matter at hand as well as the subject matter and teachings of Wright, and has confused two unrelated concepts. The governing definition of an "information stream"[11] is in the present application on page 33, lines 6 - 24, and is illustrated in Figure 7. *An information stream is entirely distinct from, and has nothing to do with computer memory allocation. The order of items in an information stream has no inherent relationship with the allocation of corresponding items in computer memory.* The term "address" with regard to information streams is defined in the present application on page 33 line 7, and is unrelated to the use of a similar term used in computer memory allocation.[11] The term "extent" is defined in the present application on page 60 line 16, and likewise has nothing to do with Wright's computer memory allocation.[11]

  - Consequently, *Wright's treatise on dynamic memory allocation is inapplicable to the subject matter of claims 6 - 7.* Wright does not teach any limitations related to information streams, and it therefore cannot be reasonably argued that Wright renders these claims as obvious.

- **Office Action Misunderstands Pratt**: In the §103(a) rejection of claims 13 - 14 (Office Action page 8), the Office Action states that "the modified Rosen, Drummond, Menezes, Cook, Muller, Gong system fails to disclose validating an output file and an execution module. However, Pratt teaches validation (see column 4 lines 22 - 32). At the time of the invention it would have been obvious to a person of ordinary skill in the art to use Pratt's validation to validate the modified Rosen, Drummond, Menezes, Cook, Muller, Gong system's output and module. Motivation to do so would have been to determine if an error has been detected which requires modification of the data (see column 4 lines 33-40)."

  - The Applicant respectfully traverses the above contention, in that the Office Action apparently misunderstands the nature of the subject matter at hand as well as the subject matter and teachings of Pratt, and has confused two unrelated concepts. The governing definition of "validation"[11] is in the present application on page 29, lines 19 - 23, and relates to the analyzing of an authentication to prove the identity of the source of authenticated information. Pratt, on the other hand, deals exclusively with the systematic checking of directory entries in a computer file system to confirm that the directory entries point correctly to the stored file data locations which they purport to represent, and to the taking of corrective action in the case of damaged directory tables. *Validation in the context of the present invention is entirely distinct from, and has nothing to do with Pratt's file systems.*

  - Consequently, *Pratt's patent on file system integrity checking is inapplicable to the subject matter of claims 13 - 14.* Pratt does not teach any limitations related to authenticated information, and it cannot be reasonably argued that Pratt renders these claims as obvious.

  - Furthermore, the motivation proposed by the Office Action for combining Pratt — "determine if an error has been detected which requires modification of the data" is insufficient to serve as a reasonable motivation, because the system to which

Pratt is proposed to be added (e.g., Rosen and Drummond) is well-known in the art to already exhibit error-detection that is more general than that of Pratt (as discussed above, with reference to Socolofsky).

- **Office Action Misunderstands Rosen**: In the §103(a) rejection of claim 5 (Office Action page 5), the Office Action states that "the {combined and} modified Rosen, Drummond, Menezes, Cook, Muller, Gong system discloses the personalization has no usage restriction associated with it (see Rosen column 4 lines 29 - 35).

  - The Applicant respectfully traverses the above contention, in that the Office action apparently misunderstands the nature of the subject matter at hand and has confused the absence of mention of a feature with an explicit disclosure of that feature's exclusion. In particular, *Rosen makes no mention whatsoever of usage restrictions or the lack thereof.* Rosen's silence on the subject of usage restrictions is because Rosen deals only with data, not with software, and data is not subject to "usage restriction" as defined in the present application (page 39 lines 5 - 10).[11]

  - Moreover, usage restrictions as defined in the present application are a characteristic feature of prior-art software protection, and none of the cited patent references in the Office Action pertain to the field of software protection, software development, or software distribution. {As shown in Appendix A, the fields of the prior art patents cited in the present Office Action consist of: an invoice and payment system (Rosen); an automated banking machine (Drummond); a transaction authorization system (Cook), and computer file system directory checking (Pratt). Rosen and Drummond, in particular, are the primary references cited in the present Office Action.}

  - The Applicant cites Moskowitz (US 5,745,569 col. 6, lines 44 - 48) in disclosing a usage restriction: "Either the user must have the extracted watermark, or the application cannot be used ... In order to extract a digital watermark, the user must have a key."

  - Thus, the exclusion of usage restrictions in the present claim 5 constitutes a novel and essential limitation, because this exclusion bars a system according to Moskowitz (for example) from infringing on claim 5. Furthermore, this claim limitation is non-obvious because it does not appear as a teaching or suggestion in the prior art.

The Applicant therefore respectfully contends that the silence of the cited references on the matter of "usage restrictions" cannot reasonably be construed as a disclosure of a system that explicitly and deliberately excludes usage restrictions (as is done in the present application), and that the present claim 5 is thereby not rendered obvious by the mere absence of any mention of usage restrictions in the references cited in the Office Action.

The Applicant also notes that the present invention teaches away from the prior art, in that prior art software protection typically involves copy-protection schemes and/or usage control schemes. The present invention teaches away from both of these techniques.

2. Lack of Reasonable Expectation of Success

As noted above, there are several cases where the present Office Action misunderstands the prior art, and this means that combinations including that prior art do not have a reasonable expectation of success.

Furthermore, where proposed combinations are vague, indefinite, and lack reasonable specificity (as presented above), there is also no reasonable expectation of success. For example, in the case where the proposed motivation of the Office Action is "to protect the user" (Office Action pages 4 - 5), there is no reasonable expectation of success because the nature of the protection and how it is to be accomplished are not specified.

In addition, there are cases where the prior art indicates that combinations proposed by the Office Action convey no reasonable expectation of success. An example of this is the proposed combination with Gong (Office Action pages 4 - 5), supposedly "to protect the user". It might be inferred that this proposed user protection would constitute some sort of security enhancement. Drummond's existing security mechanism utilizes digital signatures which are known to provide good security (Drummond, col. 14 lines 19-38). To propose introducing a new security mechanism based on a personalization created by an untried and untested combination of elements from the cited references is considered in the prior art as not having a reasonable expectation of success. Doing so is asking people who are ordinarily-skilled in the art to create a new algorithm for security purposes without benefit of the intensive high-level review process that is necessary to attain success in the field of security. As stated plainly by Schneier (pages 6 - 7), such individually-developed algorithms which have not been extensively-tested and approved through outside review by those highly-skilled in the art are not considered reliable, and have no reasonable expectation of success.

3. *Claim Limitations of the Present Invention are not Taught or Suggested by the Cited Prior Art References Individually, nor by the References in Combination*

The combination of references proposed by the Office Action does not teach or suggest all the limitations of the present claims.[8]

- Among other limitations, the present claim 1 recites limitations featuring: a software publisher; published software; delivery of published software to a customer; receipt of published software by the customer; origination of personalization by the software publisher; the authorized user of the software; notification display on a computer that the customer is the authorized user; and validation of the published software to determine origination by the software publisher. These limitations are not disclosed or suggested in any of the prior art references cited by the Office Action. *The combination of references proposed by the Office Action therefore does not teach or suggest all of the limitations of claim 1.*

- Among other limitations, the present claim 24 recites limitations featuring: deliverable published software; delivery of the published software to a customer; receipt of the published software by a customer; the authorized user of the deliverable published software; notification display on a computer that the customer is the authorized user; and validation of the published software to determine that an embedded personalization has not been altered. These limitations are not disclosed or suggested in any of the prior art references cited by the Office Action. *The combination of references proposed by the Office Action therefore does not teach or suggest all of the limitations of claim 24.*

- Among other limitations, the present claim 28 recites limitations featuring: deliverable published software; notification display on a computer that the customer is the authorized user; and validation of the published software to determine that an embedded personalization has not been altered. These limitations are not disclosed or suggested in any of the prior art references cited by the Office Action. *The combination of references proposed by the Office Action therefore does not teach or suggest all of the limitations of claim 28.*

- It is further noted that the term "customer" as used in the present claims is specifically-defined in the present application to refer to an actual or prospective user of software who has received or is intended to receive a license to use the software (page 31 lines 8 - 9). This is the governing definition of the term[11], and is distinct from the usage in Rosen and Cook (where "customer" merely denotes someone who makes a purchase from a merchant) and Drummond (where "customer" merely denotes a patron of a bank). Thus, any limitations of the present claims related to "customers" are not met by the prior art.

- Claims 1, 24, and 28 are the independent claims of the present application. Therefore, because the combination of references proposed by the Office Action does not teach or suggest all of the limitations of claims 1, 24, and 28, *the combination of references proposed by the Office Action does not teach or suggest all of the limitations of any of the present claims.*

- For example, the combined prior art references fail to suggest the claim limitation pertaining to the validation of the authenticated personalization, and the determination that the personalization, including at least the name of the customer, was originated by the software publisher and has not been altered. The combination of all the cited prior art does not disclose the concept of a software publisher, a customer for published software, limitations that pertain to the delivery of the published software to the customer, nor limitations that pertain to the receipt of the published software by the customer.

Summary

The present Office Action has rejected the claims of the present application solely on the basis of 35 USC §103(a), by alleging that the present invention is obvious in view of the prior art.

Applicant, however, has put forth in detail herein, supported by specific citations and detailed description, that the Office Action has *not* established the required[1] *prima facie* case of obviousness against the present invention. The three necessary basic criteria for such a case, as mandated in MPEP § 2143 and elsewhere[2], are not met by the Office Action. In particular, the combination of prior art proposed by the Office Action to show obviousness does not meet all the limitations of the independent claims of the present invention.[8]

In addition, the Office Action does not procedurally establish a *prima facie* case of obviousness because the rejections have not been explained with reasonable specificity.[6]

Moreover, the Applicant has cited evidence showing that the prior art teaches away from the proposed combinations and modifications of the Office Action, indicating that the combinations and modifications suggested by the Office Action do *not* have a reasonable expectation of success and therefore do *not* establish obviousness.[2]

Furthermore, the Applicant has cited evidence showing that the present invention teaches away from the prior art, clearly indicating that the present invention is *not* obvious in view of the prior art.

It is therefore respectfully submitted that claims 1 - 9, 11, 13 - 24, 26, and 28 - 33 are in condition for allowance. Notice of allowance is therefore respectfully and earnestly solicited.

Respectfully submitted,

Moshe Brody, Applicant

Date: November 22, 2005

Appendix A:

Cited Prior Art References

Prior Art References Cited in the Present Office Action

In asserting that the claims of the present Application would be obvious to a person ordinarily skilled in the art at the time the invention was made, the Office Action cites the following prior art documents:

- US Patent 6,081,790 to Rosen, "System and method for secure presentment and payment over open networks"
- US Patent 6,289,320 to Drummond, et al., "Automated banking machine apparatus and system"
- US 6,675,153 to Cook, et al., "Transaction authorization system"
- US 6,070,254 to Pratt, et al., "Advanced method for checking the integrity of node-based file systems"
- Handbook of Applied Cryptography, by Menezes, et al.
- Inside Java 2 Platform Security, by Gong
- A Survey of Programming Techniques, by Muller
- The Java Archive JAR File Format, by Sommerer
- Dynamic Data Structures, by Wright

The above references are reviewed briefly below, regarding their citations in the present Office Action.

The Rosen Patent

Rosen relates to a method for secure presentation and payment of commercial invoices over a network, using "tickets" and "remittance advice" containing various data related to the invoices and payments.

The portion of this patent cited in the Office Action is from column 4 lines 29 - 35:

> "A presentment ticket may have an Invoices and Past Due Notices Signature field **35** that is the MTA's digital signature over the invoice and/or past due notice information being presented to the customer. It may also have customer information **37** which can be acquired from the customer's credential (e.g., customer name and address, credential authority, credential expiration date)."

The above figure references **35** and **37** are from Figure 3, showing the data items making up a "presentment ticket".

Rosen treats customer personal information strictly as data and does not mention or suggest the embedding of such information in software. Rosen also does not disclose sending of software by a software publisher nor delivery of software to a customer.

According to Rosen, a "customer" is someone who owes money to a merchant in connection with a purchase of goods or services. According to Rosen, the term "authorization" refers to the granting of permission to issue payment. Rosen does not address the issue of a customer as a software user who is authorized by the software publisher to use the software, as is done in the present application.

The Drummond Patent

Drummond relates to an automated banking machine which conducts transactions and operates devices (such as a cash dispenser) within the automated banking machine in response to HTML documents and TCP/IP messages exchanged over a network with a local computer system.

The portion of this patent cited in the Office Action is from column 14 lines 19 - 38:

> "In addition, the HTML document preferably includes embedded JAVA script which has a digital signature or a means to obtain a digital signature associated with the home HTTP server **90**. The script instruction included in the document in certain embodiments causes the device application portion to access an HTTP address on a server, which in the described embodiment is server **90**. The HTTP address corresponds to an HTTP record which includes at least one instruction and preferably includes a program such as a JAVA applet or Active-X file. The instruction is used to operate the appropriate transaction function device. The HTTP record preferably includes data representative of a signature, such as a digital signature. This digital signature is received responsive to the JAVA script **82** and processed in the device application portion **84**. A JAVA applet processes the digital signature to authenticate it and if it is an acceptable signature authorizes operation of the banking machine. In certain embodiments the applet may compare the signature to signature data stored in memory for a predetermined relationship, such as a match."

The above figure references **82** and **84** are from Figure 2, showing a Java environment embedded in an automated banking machine. Figure reference **90** is from Figure 3, showing an HTTP server on a banking network.

Drummond describes how a digital signature can be used to authorize operation of a banking machine. Drummond treats authorization only in the context of using a banking machine to performi financial transactions.

According to Drummond, an authorized user is someone who is permitted to use a banking machine card, as determined by matching biometric data or knowledge of a PIN ("Personal Identification Number") [US 6289320 Drummond, col. 15 lines 45 - 48].

According to Drummond, a "customer" is someone who uses a banking machine in connection with bank account services. According to Drummond, the term "authorization" refers to the granting of permission to access the automated banking machine [US 6289320 Drummond, col. 31 lines 65].

Drummond also does not disclose sending of software by a software publisher nor delivery of software to a customer. Drummond furthermore does not address the issue of a customer as user of software who is authorized by the software publisher to use the software, as is done in the present application.

The Cook Patent

Cook relates to method and apparatus for authorizing a financial transaction between a consumer and a merchant, in which the consumer can remain anonymous to the merchant while still insuring the validity of the consumer's identity.

The only portion of this patent cited in the Office Action is from column 19 lines 13 - 22:

> "Charge slip application **118** decrypts the display authorization message. Charge slip application **118** notifies the member that the purchase was authorized **(50)**. More specifically, charge slip application **118** displays a notification to the member that can include an authorization text message, the authorization code, an exit button and a print button. The member can depress the print button **(51)**, resulting in the printing of a charge slip image for the member. The member can thereafter depress the exit button **(52)** and the transaction is complete."

The above figure reference **118** is from Figure 1 and Figure 3 showing a consumer client. Figure references **50, 51**, and **52** are from Figure 5 showing events between client applications.

According to Cook, a "customer" is someone who owes money to a merchant in connection with a purchase of goods or services (Cook usually refers to this individual as a "consumer"). According to Cook, the term "authorization" refers to the granting of permission to issue payment. Cook also does not disclose sending of software by a software publisher nor delivery of software to a customer. Cook does not address the issue of a customer as a software user who is authorized by the software publisher to use the software, as is done in the present application.

The Pratt Patent

Pratt relates to integrity-checking of the directory entries of a computer data file system in a data storage device. Directory entries must be confirmed as being consistent with the actual storage locations of the listed data files, and Pratt discloses a fast and efficient method for doing this, taking into account data storage access latency times and I/O throughput speeds.

Two excerpts from this patent are cited in the present Office Action:

Column 4 lines 22 - 32:

> "This dependency problem is circumvented by the present invention. Referring again to step **202**, the process passes to step **204**, which illustrates processing the directory entries first, to the extent possible. Each directory entry is validated in turn, and everything in each directory entry which can be validated without the corresponding F-Node is validated. The process then passes to step **206**, which depicts saving both the information from each directory entry which has not been validated and the information from each directory entry which is required to validate the corresponding F-Node."

Column 4 lines 33 - 40:

> "The process then passes within the initial cycle of the validation process to step **208**, which illustrates a determination of whether an error has been detected which requires modification of the directory entry. If so, the process proceeds to step **210**, which depicts queuing the error together with the needed corrective action until all of the F-Nodes have been processed. The process then passes to step **212**, described below."

According to Pratt, the term "validation" denotes the process of verifying that a directory entry in a data file system is consistent with the actual storage location of the listed file. Pratt does not address the issue of verifying that a block of personal information in software (a "personalization") was originated by the software publisher and has not been altered, as is done in the present application. *It is noted in the Response that the Office Action misunderstands the subject matter and teachings of Pratt.*

The Menezes Reference

Menezes is a general reference in the field of applied cryptography. The only portion of this reference cited in the present Office Action is page 361, which gives standard definitions and brief discussions of: "data integrity" and verification thereof; "data origin authentication"; and "message authentication".

The Gong Reference

Gong is a specialized reference dealing with Java platform security. The only portion of this reference cited in the present Office Action is pages 23 - 25, which deals with Java's "Basic Security Architecture".

In this section, Gong summarizes the security problem facing users regarding the risk of running downloaded software of possibly unknown origin on their computers, especially since such software may execute without the user's permission or awareness. Gong gives an overview of how Java security architecture protects the user from harm caused by untrusted software, including possibly-malicious software as well as innocent, but improperly-written software which could interfere with other Java programs.

The Muller Reference

Muller is a general reference overview of various well-known programming techniques. The only portion of this reference cited in the present Office Action is to section 2.3 regarding "modular programming".

The Sommerer Reference

Sommerer is a specialized reference dealing with the Java Archive file format. The only portion of this reference cited in the present Office Action is to an overview of the Java Archive format and the features and benefits thereof, including built-in authentication and validation security and compression features.

The Wright Reference

Wright is a general reference which briefly discusses the properties and features of data structures which are allocated in memory during program runtime rather than at compile or link time. The present Office Action cites this reference to establish data storage can be dynamically allocated in computer memory at runtime (page 7). *It is noted in the Response that the Office Action misunderstands the subject matter and teachings of Wright.*

Prior Art References Cited in the Present Response

- Applied Cryptography, by Schneier
- RFC 1180 — A TCP/IP Tutorial, by Socolofsky, et al.
- Transport Control Protocol, Ahonen, et al.
- US 5,745,569 to Moskowitz, et al., "Method for stega-cipher protection of computer code"

The Applicant also makes reference to the documents cited in the present Office Action, as previously listed. The above references are reviewed briefly below, regarding their citations in the present response:

The Schneier Reference

Schneier is a general reference in the field of applied cryptography. The portion of this reference explicitly cited in the present response is from pages 6 - 7 (attached).

Schneier teaches that established, widely-reviewed, and well-tested methods provide superior protection and security. There is virtually no expectation of success when innovating methods that are not subjected to intense and far-reaching scrutiny.

The Socolofsky Reference

Socolofsky is a general high-level tutorial covering basic TCP/IP and related Internet protocols encompassed thereby, as were well-established as of the time the present invention was made. Socolofsky presents an overview of the properties of networks based upon this technology. The Applicant cites Socolofsky as representative of the prior art and its teachings as relate to Rosen, Drummond, and Cook, all of which rely explicitly on an environment based on the Internet or on Internet technology (both of which are described by Socolofsky):

> *Rosen:* "The present invention describes a system that enables a merchant to securely present invoices or past due notices to a customer over an open network like the internet." [US 6081790 Rosen, col. 1 lines 47 - 49]
>
> *Drummond:* "Messages in wide area networks may be communicated using the Transmission Control Protocol/Internet protocol ('TCP/IP')." [US 6289320 Drummond, col. 2 lines 40 - 42] "Thus there exists a need for an automated banking machine and system that can be used in a wide area network such as the Internet while providing a high level of security." [US 6289320 Drummond, col. 2 lines 62 - 65]

> *Cook:* "A distributed real-time software application (referred to herein as 'ZixCharge') is provided that allows consumers to authorize transactions in a secure, private, and convenient manner for the purchase of goods and services over the Internet." [US 6675153 Cook, col. 1 lines 52 - 56]

As per a network "like the internet", this term would encompass networks based on internet technology, as Socolofsky notes:

> "The generic term 'TCP/IP' usually means anything and everything related to the specific protocols of TCP and IP. It can include other protocols, applications, and even the network medium. ... A more accurate term is 'internet technology'. A network that uses internet technology is called an 'internet'. [Sokolofsky, page 2]
>
> Pages 1 - 2 are cited in the present response (attached).

The Ahonen Reference

Ahonen is a general reference to the application of TCP in networks, presented herein as a companion prior-art document to Socolofsky. Ahonen relates TCP (the Internet "Transport Control Protocol") to the transport layer of the ISO OSI network model. The only portion of this reference explicitly cited in the present response is from pages 5 and 7 - 13 (attached), which teach that TCP is a reliable data transport mechanism that uses checksums to maintain data integrity and handshaking to make sure transmitted data is received intact.

The Moskowitz Patent

Moskowitz has already been referenced extensively in this case, particularly in the Office Action mailed October 20, 2004, where the Examiner cited Moskowitz. In the present response, the Applicant calls attention to Moskowitz as being typical of the prior art in the field of software protection, in that Moskowitz teaches placing usage controls and/or restrictions on the software as an essential feature of the protection. The nature of these usage controls and/or restrictions is detailed in the Applicant's response to the Office Action mailed October 20, 2004.

Except for Moskowitz, which is already included in the file history of the present application, the relevant portions of the above references are appended to the present response.

Appendix B:

Overview of the Present Invention Elements and Inventor's Motivations for Combining Them

For the convenience of the Examiner, and to clarify the elements of the present invention and the motivations for combining them, and to contrast these elements and motivations with those of the obviousness argument in the present Office Action, the Applicant provides the following overview of the present invention:

- The field is in the area of protecting the intellectual property rights of software publishers against unauthorized distribution of copies of the software — particularly against "casual" copying and distribution by the customers themselves. The intended benefit is principally for the software publishers. The problem addressed by the present invention is how to provide protection that meets the following criteria:

  - The means of protection must be applicable to network distribution of software, where the software is supplied to the customer (the authorized user) as a pure information stream without any connection to physical media.

  - The means of protection must avoid all possibility of customer dissatisfaction that might result from added complexity of installation or use, customer inconvenience, interference with the rights of the customer to use the software in any legitimate way, or risk of operational malfunction.

  - For example, the customer must never have to perform any inconvenient steps during installation or setup (such as entering a software key); the customer must never be restricted in any legitimate use of the software (such as restricting the use of the software to a single computer); and there must be absolutely no chance that the protective method will erroneously restrict operation of the software (such as by disabling the software when the computer is re-configured, because the protection interprets the reconfigured computer as a different computer from the one on which the software was installed).

  - The present inventor (the Applicant) realized that customer dissatisfaction with active software protection was the principal factor in the commercial failure of many software protection schemes, and that whatever benefits a software publisher might receive from active software protection could be outweighed by the losses stemming from consumer rejection. Thus, the inventor's motivation for adopting the purely passive software protection of the present invention was to avoid this hazard.

- The present inventor (the Applicant) observed that customers tend to be reluctant to distribute copies of software with which they are personally associated and for which they are responsible, such as software which incorporates their name and other personal information. Therefore, a method of incorporating the customer's name and other personalized information into the software and prominently displaying this during use of the software might be able to provide the desired protection.

- The present inventor was aware that much current software is designed to be configured by the customer at the time of installation, and that the installation process often includes having the customer enter personalized information, which is embedded into the installed software on the customer's computer at the time of installation, and which may subsequently be displayed by the software. However, the prior art relies on the customer to input this information; the fact that the insertion of this information is manually performed by the customer on the customer's own computer excludes the use of strong cryptographic methods to provide true security for the embedded personalized information. Cryptographic keys, if used at all by the prior art, must be short and simple, and the keys for encryption and decryption are necessarily available to a determined attacker. The result is that prior art personalized information embedded in the software is insecure and easily compromised. As a result, prior art methods for embedding personalized information of the authorized user are of little or no value in protecting the software from unauthorized copying and distribution.

- The present inventor realized, however, that a growing trend in software distribution was via networks, such as the Internet. The present inventor also realized that when distributing software via a network, each instance of software is separately downloaded by a server to the particular customer. The present inventor thus realized that in the context of network distribution, it is possible for the software publisher to separately embed each customer's personalized information into the software after receiving the order and before delivering the software.

- Therefore, the present inventor developed a method by which a customer's personal information, including the customer's name, would be embedded in a secure manner within the individual copy of the software before being sent to the customer. The present inventor also provided that the customer's name would be displayed during software operation with a notification that the identified customer is the authorized user of the software. The motivation for this was to associate the authorized user's name and other personal information with the software, and to make the customer aware of this, thereby discouraging the customer from distributing copies of the software to others, in keeping with the points above.

- The present inventor provided that the personalized information embedded in the software would be protected by strong cryptographic methods, which are feasible to use in the software publisher's environment. The motivation for this was to prevent the embedded information from being altered or removed by the customer or other attacker.

- The present inventor also provided for authentication and validation of the personalized information of the authorized user. Authentication and validation were to be done by strong cryptographic methods. The motivation for this was to allow the software to verify that the personalized information was originated by the software publisher and had not been altered or removed by the customer or other attacker.

- Essential elements of the present invention include:

  - Embedding of the customer's personal information (the "personalization"), including the customer's name, by the software publisher into the software after receiving the order and before delivering the software to the customer.

  - The use of strong encryption ("authentication") prior to delivery of the software to the customer, to secure the personalization against change by the customer or other attacker.

  - The use of strong encryption ("validation") during software operation after receipt of the software by the customer, to verify that the personalization is from the software publisher and has not been modified.

  - Display of the customer's validated personal information, including the customer's name, during software operation to inform the user that the identified customer is the authorized user of the software.

- The present invention, as embodied in the above elements, and as claimed in the present application, represents an innovative and novel solution to the problem of protecting the intellectual property rights of the software publisher.

- In addition, the present inventor also provided that the protective measures of the present invention explicitly not be associated with any usage control or usage restrictions on the software (including restrictions on copying the software). This is in keeping with the criteria above, which requires that the protection must avoid all possibility of user dissatisfaction. Thus, the protection afforded by the present invention is passive, to prevent any possible interference with the use of the software.

- *This aspect of the present invention is not only novel, but is also* *non-obvious*, in that the prior art consistently teaches away from such a provision. The prior-art regarding the protection of software from unauthorized copying and distribution overwhelmingly teaches placing active controls and restrictions on the use of software.

The above overview illustrates how the elements of the present invention and the motivations for using them are unified by the goal of the invention, and how the elements of the present invention are considered as a whole.

The present Application relates the above information in detail and elaborates on these concepts in the description and drawings.

Applied Cryptography

Protocols, Algorithms, and Source Code in C

Bruce Schneier

John Wiley & Sons, Inc.
NEW YORK • CHICHESTER • BRISBANE • TORONTO • SINGAPORE

Associate Publisher: Katherine Schowalter
Editor: Paul Farrell
Managing Editor: Beth Austin
Editorial Production & Design: Editorial Services of New England, Inc.

This publication is designed to provide accurate and authoritative information in regard to the subject matter covered. It is sold with the understanding that the publisher is not engaged in rendering professional services. If legal, accounting, medical, psychological, or any other expert assistance is required, the services of a competent professional person should be sought. ADAPTED FROM A DECLARATIONS OF PRINCIPLES OF A JOINT COMMITTEE OF THE AMERICAN BAR ASSOCIATION AND PUBLISHERS.

In no event will the publisher or author be liable for any consequential, incidental, or indirect damages (including damages for loss of business profits, business interruption, loss of business information, and the like) arising from the use or inability to use the protocols and algorithms in this book, even if the publisher or author has been advised of the possibility of such damages.

Some of the protocols and algorithms in this book are protected by patents and copyrights. It is the responsibility of the reader to obtain all necessary patent and copyright licenses before implementing in software any protocol or algorithm in this book. This book does not contain an exhaustive list of all applicable patents and copyrights.

Some of the protocols and algorithms in this book are regulated under the United States Department of State International Traffic in Arms Regulations. It is the responsibility of the reader to obtain all necessary export licenses before implementing in software for export any protocol or algorithm in this book.

This text is printed on acid-free paper.

Trademarks
Designations used by companies to distinguish their products are often claimed as trademarks. In all instances where John Wiley & Sons, Inc. is aware of a claim, the product names appear in initial capital or all capital letters. Readers, however, should contact the appropriate companies for more complete information regarding trademarks and registration.

Library of Congress Cataloging-in-Publication Data
Schneier, Bruce
Applied cryptography : protocols, algorithms, and source code in C
/ Bruce Schneier.
p. cm.
Includes bibliographical references and index.
ISBN 0-471-59756-2 (paper)
1. Computer security. 2. Telecommunication—security measures. 3. Cryptography. 4. Title

QA76.9.A25S35 1993
005.8'2—dc20 93-2139
CIP

Printed in the United States of America
10 9 8 7 6 5 4

when cryptanalysts have a tamperproof box that does automatic decryption, the job is to deduce the key.

Given: $C_1, P_1 = D_k(C_1), C_2, P_2 = D_k(C_2), \ldots C_i, P_i = D_k(C_i)$
Deduce: k

This attack is primarily applicable to public-key cryptosystems and will be discussed in Section 12.4. A chosen-ciphertext attack also works against a symmetric algorithm, but due to the symmetry of these cryptosystems it is equivalent in complexity to a chosen-plaintext attack.

6. **Chosen-key attack.** This is not an attack when you're given the key. It's strange and obscure, not very practical, and is discussed in Section 10.1.

Known-plaintext attacks and chosen-plaintext attacks are more common than you might think. It is not unheard of for a cryptanalyst to get a plaintext message that has been encrypted or to bribe someone to encrypt a chosen message. You may not even have to bribe someone; if you give a message to an ambassador, you will probably find that it gets encrypted and sent back to the United States for consideration. Many messages have standard beginnings and endings that might be known to the cryptanalyst. Encrypted source code is especially vulnerable because of the regular appearance of keywords: `#define`, `struct`, `else`, `return`. Encrypted executable code has the same kinds of problems, called protocols, loop structures, etc. David Kahn's books [462,463,464] have some historical examples of these kinds of attacks.

One of the fundamental axioms of cryptography is that the enemy is in full possession of the details of the algorithm and lacks only the specific key used in the encryption. (Of course, one would assume that the CIA does not make a habit of telling Mossad about its cryptographic algorithms, but Mossad probably finds out anyway.) While this is not always true in real-world cryptanalysis, it is always true in academic cryptanalysis; and it's a good assumption to make in real-world cryptanalysis. If others can't break an algorithm even with knowledge of how it works, then they certainly won't be able to break it without that knowledge.

Cryptanalysts don't always have access to the algorithm—for example, when the United States broke the Japanese diplomatic code, PURPLE, during World War II [462]—but they often do. If the algorithm is being used in a commercial security program, it is simply a matter of time and money to disassemble the program and recover the algorithm. If the algorithm is being used in a military communications system, it is simply a matter of time and money to buy (or steal) the equipment and reverse-engineer the algorithm. There have been many historical instances when cryptanalysts did not know the encryption algorithms; sometimes they broke the algorithm anyway, and sometimes they did not. In any case, it is unrealistic to rely on it.

Those who claim to have an unbreakable cipher simply because they can't break it are either geniuses or fools. Unfortunately, there are more of the latter in the world. Beware of people who extol the virtues of their algorithms, but refuse to make them public; trusting their algorithms is like trusting snake oil.

On the other hand, a good algorithm can be made public without worry. You can send it to your adversaries, publish it in a magazine, or shout it from the rooftops. It doesn't matter; even the designer of the algorithm can't decrypt messages without the key.

Good cryptographers rely on peer review to separate the good algorithms from the bad.

Security of Cryptosystems

Different cryptosystems have different levels of security, depending on how hard they are to break. As we will see, all algorithms but one are theoretically **breakable,** given enough time and computing resources. If the time and money required to break an algorithm is more than the value of the encrypted data, then it is probably safe. Computers are becoming increasingly faster and cheaper. At the same time, the value of the data decreases over time. It is important that those two lines never cross.

Some algorithms are only breakable with the benefit of more time than the universe has been in existence and a computer larger than all the matter in the universe. These algorithms are theoretically breakable, but not breakable in practice. An algorithm that is not breakable in practice is **secure.**

An algorithm is **unconditionally secure** if, no matter how much ciphertext a cryptanalyst has, there is not enough information to recover the plaintext. In point of fact, only a one-time pad (see Section 1.2.4) is unbreakable given infinite resources. Cryptography is more concerned with cryptosystems that are computationally unfeasible to break. An algorithm is considered **computationally secure,** or **strong,** if it cannot be broken with available (current or future) resources. Exactly what constitutes "available resources" is open to interpretation.

The amount of computing time and power required to recover the encryption key is called the **work factor,** and is expressed as an order of magnitude. If an algorithm has a work factor of 2^{128}, then 2^{128} operations are required to break the algorithm. These operations can be very complex and time-consuming, but details of the operations are left to the implementation. Still, if you assume that you have enough computing speed to perform a million of them every second, and you set a million parallel processors against the task, it will still take over 10^{19} years to recover the key. (For comparison's sake, the age of the universe is estimated at 10^{10} years.) I would consider an algorithm that takes a billion times the age of the universe to break to be computationally secure.

While the work factor required to break a given algorithm is constant (that is, until some cryptanalyst finds a better cryptanalytic attack), computing power is anything but. During the last half-century we have seen phenomenal advances in computing power, and there is no reason to think that it will change anytime soon. Many cryptanalytic attacks are perfect for parallel machines: the task can be broken down into billions of tiny pieces and none of the processors needs to interact with another. Pronouncing that an algorithm is secure simply because it is unfeasible to break, given current technology, is dicey at best. Good cryptosystems are designed to be unfeasible to break with the computing power that is expected to evolve many years in the future.

Network Working Group
Request for Comments: 1180

T. Socolofsky
C. Kale
Spider Systems Limited

January 1991

A TCP/IP Tutorial

Status of this Memo

This RFC is a tutorial on the TCP/IP protocol suite, focusing particularly on the steps in forwarding an IP datagram from source host to destination host through a router. It does not specify an Internet standard. Distribution of this memo is unlimited.

Table of Contents

1. Introduction

This tutorial contains only one view of the salient points of TCP/IP, and therefore it is the "bare bones" of TCP/IP technology. It omits the history of development and funding, the business case for its use, and its future as compared to ISO OSI. Indeed, a great deal of technical information is also omitted. What remains is a minimum of information that must be understood by the professional working in a TCP/IP environment. These professionals include the systems administrator, the systems programmer, and the network manager.

This tutorial uses examples from the UNIX TCP/IP environment, however the main points apply across all implementations of TCP/IP.

Note that the purpose of this memo is explanation, not definition. If any question arises about the correct specification of a protocol, please refer to the actual standards defining RFC.

The next section is an overview of TCP/IP, followed by detailed descriptions of individual components.

2. TCP/IP Overview

The generic term "TCP/IP" usually means anything and everything related to the specific protocols of TCP and IP. It can include other protocols, applications, and even the network medium. A sample of these protocols are: UDP, ARP, and ICMP. A sample of these applications are: TELNET, FTP, and rcp. A more accurate term is "internet technology". A network that uses internet technology is called an "internet".

2.1 Basic Structure

To understand this technology you must first understand the following logical structure:

```
                  ----------------------------
                  |    network applications  |
                  |                          |
                  |...  \ | /  ..  \ | /  ...|
                  |     -----      -----     |
                  |     |TCP|      |UDP|     |
                  |     -----      -----     |
                  |         \      /         |
                  |         --------         |
                  |         |  IP  |         |
                  |  -----  -*------         |
                  |  |ARP|   |               |
                  |  -----   |               |
                  |      \   |               |
                  |      ------              |
                  |      |ENET|              |
                  |      ---@--              |
                  ----------|-----------------
                            |
      ----------------------o---------
          Ethernet Cable
```

Figure 1. Basic TCP/IP Network Node

This is the logical structure of the layered protocols inside a computer on an internet. Each computer that can communicate using internet technology has such a logical structure. It is this logical structure that determines the behavior of the computer on the internet. The boxes represent processing of the data as it passes through the computer, and the lines connecting boxes show the path of

TRANSPORT CONTROL PROTOCOL

October 19th, 1998

Kimmo Ahonen
Kimmo.Z.Ahonen@ntc.nokia.com

Juha Koskelainen
Juha.Koskelainen@ntc.nokia.com

Department of Computer Science
University of Helsinki

Abstract

Aim of this paper is to present transmission control protocol of Internet (TCP) and demonstrates some of the problems in modern networks while using TCP.

2. The Data Link layer builds on the transmission capability of the Physical layer and provides services to the Network layer. The bits that are transmitted or received are grouped in logical units called a frame. In the context of LANs, a frame could be a Token Ring or Ethernet frame, an FDDI (Fiber Distributed Data Interface) frame, or another LAN type frame. For WAN links, this could be a SLIP (Serial Line Interface Protocol), PPP (Point-to-Point Protocol), X. 25, or an ATM (Asynchronous Transfer Mode) cell frame or another WAN type frame [3].

3. The Network layer builds on the node-to-node connection provided by the Data Link layer. Network layer provides additional service how to route packets (units of information at the network layer) between nodes connected through an arbitrarily complex network.

Besides routing, the Network layer helps to eliminate congestion as well as regulate the flow of data. The Network layer also makes it possible for two networks to be interconnected by implementing a uniform addressing mechanism [3].

4. The Transport layer provides enhancements to the services of the Network layer. This layer helps ensure reliable data delivery and end-to-end data integrity. To ensure reliable delivery, the transport layer builds on the error control provided by the lower levels [3]. If the lower layers cannot provide error free data flow, the Transport layer has to resolve data errors.

5. The Session layer allows two applications on different computers to establish, use, and end a connection called a session. The layer performs name recognition and the functions needed to allow two applications to communicate over the network, such as security functions.

6. The Presentation layer managers the way data is represented. Many ways of representing data exist, such as ASCII and EBCDIC for text files. Many TCP/IP applications do not use any Presentation layer services and the Presentation layer is null for these applications [3].

7. The Application layer contains the protocols and functions needed by user applications to perform communication tasks. These protocols provide different services for user applications to interact with computer network.

Many of these services are called Application Programming Interfaces (APIs). APIs are programming libraries that an application writer can use to write network applications.

Two computer systems involved in the data exchange are called End Systems (ES) in OSI terminology. The application layer in each ES performs processing and adds some application information as header information to the message. In every layer some more processing information are added to the to the message. The following picture shows an example of how data traverses the network.

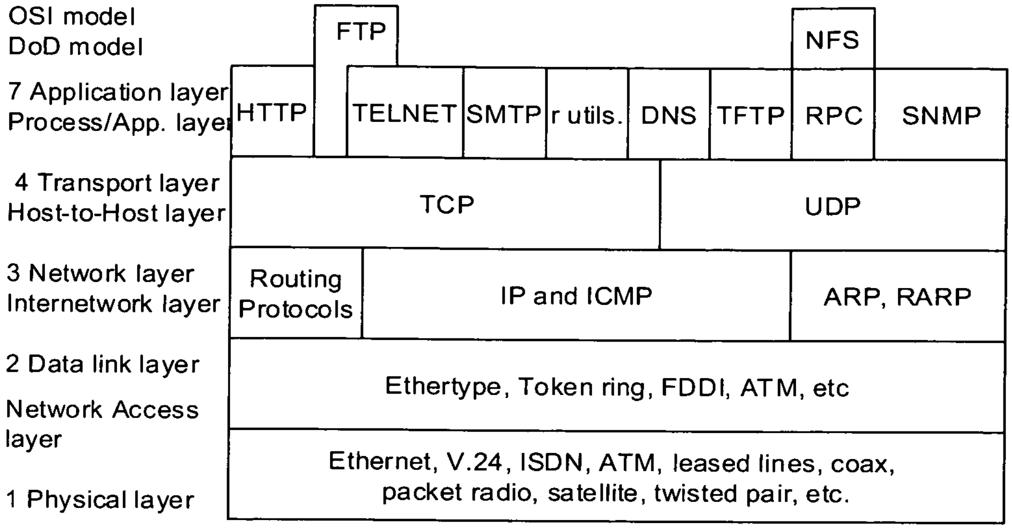

Figure 2: TCP/IP implementation hierarchy

Different applications use services provided by TCP, UDP and IP. Even though TCP and UDP are higher level protocols than IP applications can use IP directly. Some examples of TCP level applications are File Transfer Protocol (FTP), Hypertext Transfer Protocol (HTTP), TELNET and Simple Mail Transfer Protocol (SMTP). Examples of protocols that run on the top of UDP are Trivial File Transfer Protocol (TFTP) and Simple Network Management Protocol (SNMP). Network File System (NFS) is an example of protocol that can run on either TCP or UDP.

TCP/IP can run on top of many lower level protocols.

The Transmission Control Protocol (TCP)

The TCP protocol provides a standard general-purpose method for reliable delivery of data. For applications TCP provides a standard way of accessing remote computers on unreliable internetwork. This reliability is provided by adding services on top of IP. IP is connectionless and does not guarantee delivery of packets.

The reliability of TCP is achieved by retransmitting data, which has been sent but not acknowledged by receiver within given time. Thus sending TCP must keep the sent data in memory until it has received the acknowledgements of sent data.

TCP assumes that IP is inherently unreliable, so TCP adds services to ensure end-to-end delivery of data. TCP has very few expectations on the services provided by the networks and it thus can be run across a large variety of hardware. All that is required from lower level is unreliable datagram service.

TCP is the primary transport protocol used to provide reliable, full-duplex, virtual circuit connections. The most common use of TCP is to run it over IPv4 or IPv6, although several experimental projects have been done to run TCP on other Network layer protocols [4].

IP is implemented on hosts and routers. TCP is typically implemented on hosts only. Today, many routers are implemented with TCP protocol to provide easy configuration and management. For example, many commercial routers implement TCP or UDP to provide remote login and network management facilities. Even though TCP and UDP are implemented in routers, the transport protocols are not used by routing services and messages. This is illustrated in the following picture.

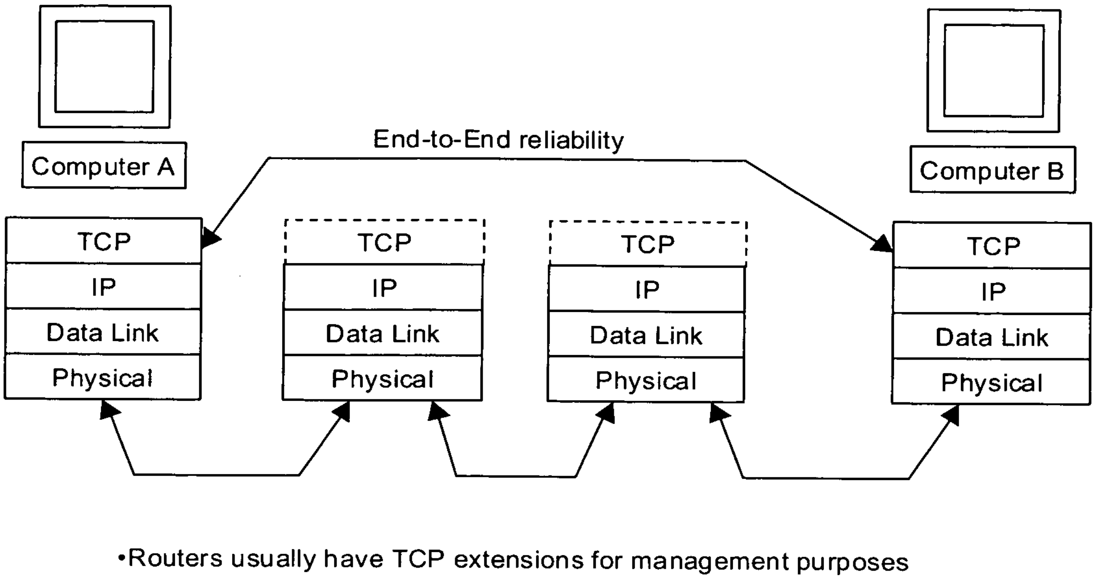

Figure 3: TCP protocol on network

TCP Standard

TCP standard is defined in RFC 793 in 1981. The primary purpose of the TCP is to provide reliable, securable logical circuit or connection service between pairs of processes [4]. This security is based on assumption that the underlying network can be trusted, which is not the case in the current commercial Internet. The statement secure comes from the time, when TCP/IP was primarily used for Military purposes.

TCP provides reliable services on top of a less reliable internet communication systems on following areas [4]:

- Basic Data Transfer
- Reliability
- Flow Control
- Multiplexing
- Connections
- Precedence and Security

The basic operation of the TCP is described in following sections [4].

The Basic Data Transfer

The TCP Basic Data Transfer is able to transfer a continuous stream of octets in each direction between its users by packaging some number of octets into segments for transmission through the internet system. The octets are sent among application processes running on remote systems that use TCP. In general, the TCPs decide when to block and forward data at their own convenience [4].

The application processes group a set of bytes that need to be sent or received into a message segment. Message segments can be arbitrary length. At the TCP level there is no real restriction on message size because the details of accommodating the message segments in IP datagrams is the task of the IP layer.

Ultimately, the messages have to be sent in IP datagrams that are limited by the MTU (Maximum Transfer Unit) size of a network interface. For efficiency reasons TCP connections typically negotiate a maximum segment size.

Messages sent by TCP have an octet orientation. TCP keeps track of octets that has been sent or received. The TCP does not have any notion of a block of data. This differs from many other transport protocols, which typically keep track of the Transport Protocol Data Unit (TPDU) number and the octet number. TCP can be used to provide multiple connections between two host computers.

Application processes are allowed to send data whatever size that is convenient for sending. For example, an application can send one octet at a time or several kilo octets. TCP numbers each octet that is send. The octets are delivered to the application layer in same order that they are sent.

An application can send data to TCP a few octets at a time. TCP buffers this data and sends these octets either as a single message or as several smaller message segments. All that TCP guarantees is that data arrives in the order in which it was sent.

The actual data that is sent by TCP is treated as an unstructured stream of octets. TCP does not contain any facility to superimpose an application dependent structure on data. The structuring of data must be handled by the application processes that communicate by using TCP.

Reliability

The TCP must recover from data that is damaged, lost, duplicated, or delivered out of order by the internet communication system. This is achieved by assigning a sequence number to each octet transmitted, and requiring a positive acknowledgment (ACK) from the receiving TCP. If the ACK is not received within the timeout interval, the data is retransmitted. At the receiver, the sequence numbers are used to correctly order segments that may be received out of order. Damaged segments are handled by adding a checksum to each segment transmitted. The receiver verifies the checksum discarding damaged segments. Unless there is a physical break in the link that causes physical partitioning of the network, TCP is able to recover most internet communications system errors.

Flow Control

TCP provides a means for the receiver to govern the amount of data sent by the sender. Computers that send and receive TCP data segments can operate at different data rates because of differences in CPU and network bandwidth. As a result, it is possible for sender to send data at a faster rate than the receiver can handle.

TCP implements a flow control mechanism that controls the amount of data send by the sender. This is achieved by using a sliding window mechanism. The receiver TCP module sends back to the sender an acknowledgment that indicates a range of acceptable sequence numbers beyond the last successfully received segment. This range of acceptable sequence numbers is called a window.

The window size reflects the amount of buffer space available for new data at the receiver. If this buffer space size shrinks because the receiver is being overrun, the receiver will send back a smaller window size. In the extreme case the windows size will decrease to very small or one octet. This is referred to as the silly window syndrome. Most TCP implementations take special measure to avoid it.

The goal of the sliding window mechanism is to keep the channel full of data and to reduce the delays for waiting acknowledgements.

Multiplexing

TCP enables many processes within a single host computer to use TCP communications simultaneously. Different processes may be communicating over the same network interface. Thus they must be separated from each other. This separation is done by using different port numbers for each process. Port numbers are concatenated with network and host addresses from the internet communication layer, this forms a socket.

A pair of sockets uniquely identifies a connection. Multiple connections can be used to enable several connections between application processes on remote computers. The binding of ports to processes is handled independently by each computer. Frequently used processes are attached to fixed sockets, which are made known to the public.

Connections

The reliability and flow control mechanisms require that TCPs initialize and maintain status information for data streams. The combination of the sockets, sequence numbers and window sizes is called a connection. Each connection is uniquely specified by a pair of sockets identifying its two sides.

The TCP connection is identified by the parameters of both end points:

(IP address 1, port number 1, IP address 2, port number 2)

These parameters make it possible to have several application processes that connect to the same remote end point [4].

Port number is a 16-bit value. This means that port numbers can vary in the range of 0 to 65535. Some of these port numbers are listed on the following table [3, 5]:

Table 1: Port numbers

Protocol	Number	Application Layer Service
TCP/UDP	0	
TCP/UDP	7	Echo
TCP/UDP	9	Discard
TCP	17	Quote of the Day (QUOTD)
TCP	20	FTP Data Port
TCP	21	FTP Control Port
TCP	22	SSH - Secure Shell
TCP	23	Telnet
TCP	25	SMTP
TCP/UDP	53	Domain Name Server (Domain)
UDP	67	Bootstrap Protocol Server (bootps)
UDP	68	Bootstrap Protocol Client (bootpc)

UDP	69	Trivial Transfer Protocol (tftp)
TCP/UDP*	79	Finger protocol
TCP/UDP*	80	HTTP Hyper Text Transfer protocol (World Wide Web)
TCP/UDP*	110	Post Office Protocol - Version 3 (POP3)
TCP/UDP*	137	NETBIOS Naming Service (netbios-ns)
TCP/UDP*	138	NETBIOS Datagram Service (netbios-dgm)
TCP/UDP*	139	NETBIOS Session Service (netbios-ssn)
UDP	161	Simple Network Management Protocol (SNMP)
TCP	443	HTTPS - HTTP over SSL/TLS
TCP	513	Remote Login
TCP	515	LPR/LPR printing
UDP	1512	Microsoft WINS
TCP	1525	Oracle SGL*net v1
TCP	6000-6063	X11 protocol

*Teleware training material [5] specifies this as TCP only. Inside TCP/IP book specifies this as both TCP/UDP.

Port numbers on the range 0..1023 are called well-known port numbers. Many publicly available TCP/IP applications use port numbers on this well-known range.

Precedence and security

RFC 793, where TCP was originally specified, states precedence and security:

The users of TCP may indicate the security and precedence of their communication. Provision is made for default values to be use when these features are not needed [4].

This assumption was made when the network was assumed to be secure. In these days the network cannot be trusted. Internet messages can be read by virtually anybody.

TCP operations

TCP is implemented as a protocol module that interacts with the computer's operating system. In many operating systems, the TCP module is accessed like the file system of the operating system. The TCP module depends on other operating system functions to manage its data structures and services. The interface to the physical network is controlled by a device driver module. TCP does not communicate directly to device driver. IP module acts as a middle layer in TCP communication to the network driver.

From the abstract viewpoint, applications will interface with the TCP module with the following system calls [4]:

OPEN to open a connection

CLOSE to close a connection

SEND to send data to an open connection

RECEIVE to receive data from an open connection

STATUS to find information about a connection

These calls are much like operating system's file system calls. The connection must be established before it can be used, as is with operating system files.

TCP Message Format

TCP segments are sent as internet datagrams. The Internet Protocol header carries several information fields, including the source and destination host addresses [4]. A TCP header follows the internet header, supplying information specific to the TCP protocol. This division allows for the existence of host level protocols other than TCP.

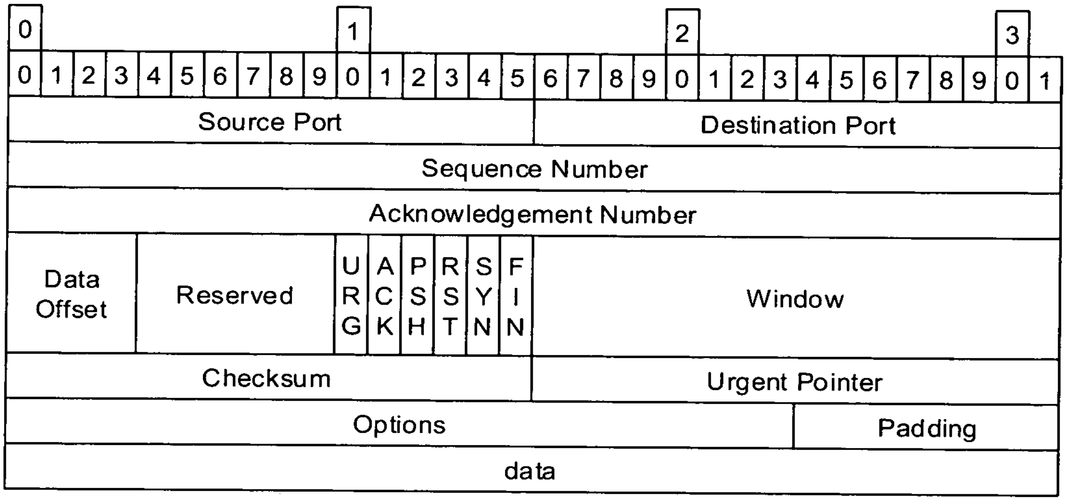

Figure 4 TCP header format

Table 2 TCP header specification

Source port:	16 bits	The source port number
Destination port:	16 bits	The destination port number
Sequence Number (SEQ):	32 bits	The sequence number of the first data octet in this segment (except when SYN is present) If SYN is present the sequence number is the initial sequence number (ISN) and the first data octet is ISN+1
Acknowledgement Number (ACQ):	32 bits	If the ACK control bit is set this field contains the value of the next sequence number the sender of the segment is expecting to receive. Once a connection is established this is always sent.
Data Offset:	4 bits	The number of 32 bit words in the TCP header. This indicates where the data begins. The TCP header (even one including options) is an integral number of 32 bits long.
Reserved:	6 bits	Reserved for future use. Must be zero.
Control bits:	6 bits (from left to right)	
	URG:	Urgent Pointer field significant
	ACK:	Acknowledgement
	PSH:	Push function
	RST:	Reset the connection
	SYN:	Synchronize sequence numbers
	FIN:	No more data from sender
Window:	16 bits	The number of data octets beginning with the one indicated in the acknowledgement field which the sender of this segment is willing to accept.
Checksum:	16 bits	Checksum field is calculated to verify the data correctness.

A TCP connection progresses from one state to another in response to events. The events are the user calls, OPEN, SEND, RECEIVE, CLOSE, ABORT, and STATUS; the incoming segments, particularly those containing the SYN, ACK, RST and FIN flags; and timeouts [4].